

(43) Date of A Publication **04.09.2002**

(21) Application No 0104955.0

(22) Date of Filing **28.02.2001**

(71) Applicant(s)
NTL Group Ltd
(Incorporated in the United Kingdom)
NTL House, Bartley Wood Business Park, Hook,
Hamshire. RG27 9XA, United Kingdom

(72) Inventor(s)
Andrew Arthur Victor Wolffsohn
Paul Richard Hill

(74) Agent and/or Address for Service
Marks & Clerk
57-60 Lincoln's Inn Fields, LONDON, WC2A 3LS,
United Kingdom

(51) INT CL⁷
H04B 17/00 // H04H 1/02 , H04N 17/00

(52) UK CL (Edition T)
H2K KHF KJG KSC2 K490

(56) Documents Cited
EP 1014279 A1
US 4752927 A
WO 2000/038358 A2

(58) Field of Search
UK CL (Edition S) H2K KHF KJBC KJS KSC2 , H4P
PEM
INT CL⁷ H04B 17/00 , H04H 1/02 , H04N 17/00
ONLINE : WPI EPODOC JAPIO

(54) Abstract Title

Adaptive fault detection and localisation in television distribution networks using digital signal processing

(57) Faults in one or more signal channels are detected and evaluated, preferably faults in audio input channels that form part of a cable TV distribution network. The system comprises an error detector receiving an input channel and outputting an error signal to a fault evaluator. There may be multiple input channels and multiple error detectors, or many input channels may be multiplexed to one error detector. The fault evaluator may have fault specification data corresponding to the input channel type or signal type. The fault evaluator may indicate which input channel or signal has a fault. The fault detection system may be realised using digital signals and by running program code on a processor. The fault detectors and fault evaluators may communicate across a data network with each other and a central server. Audio faults detected may include silence, drop-outs, stereo channel imbalance and tone faults.

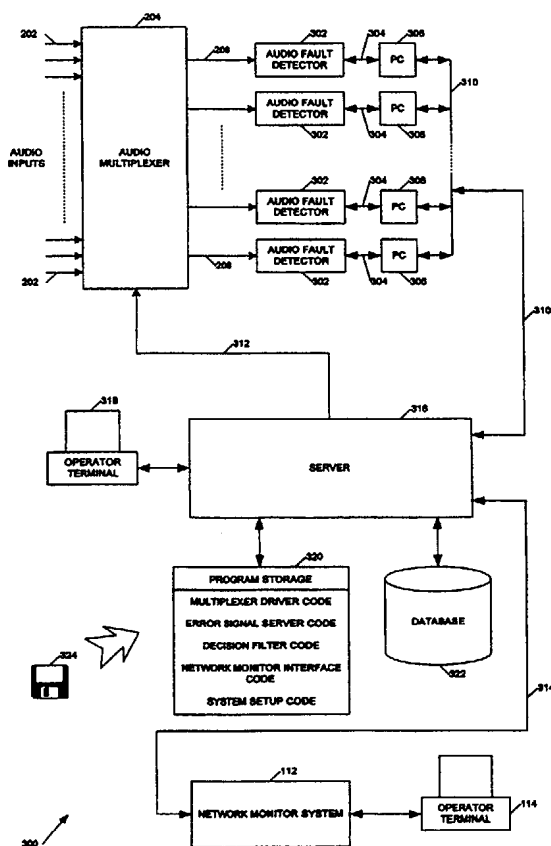
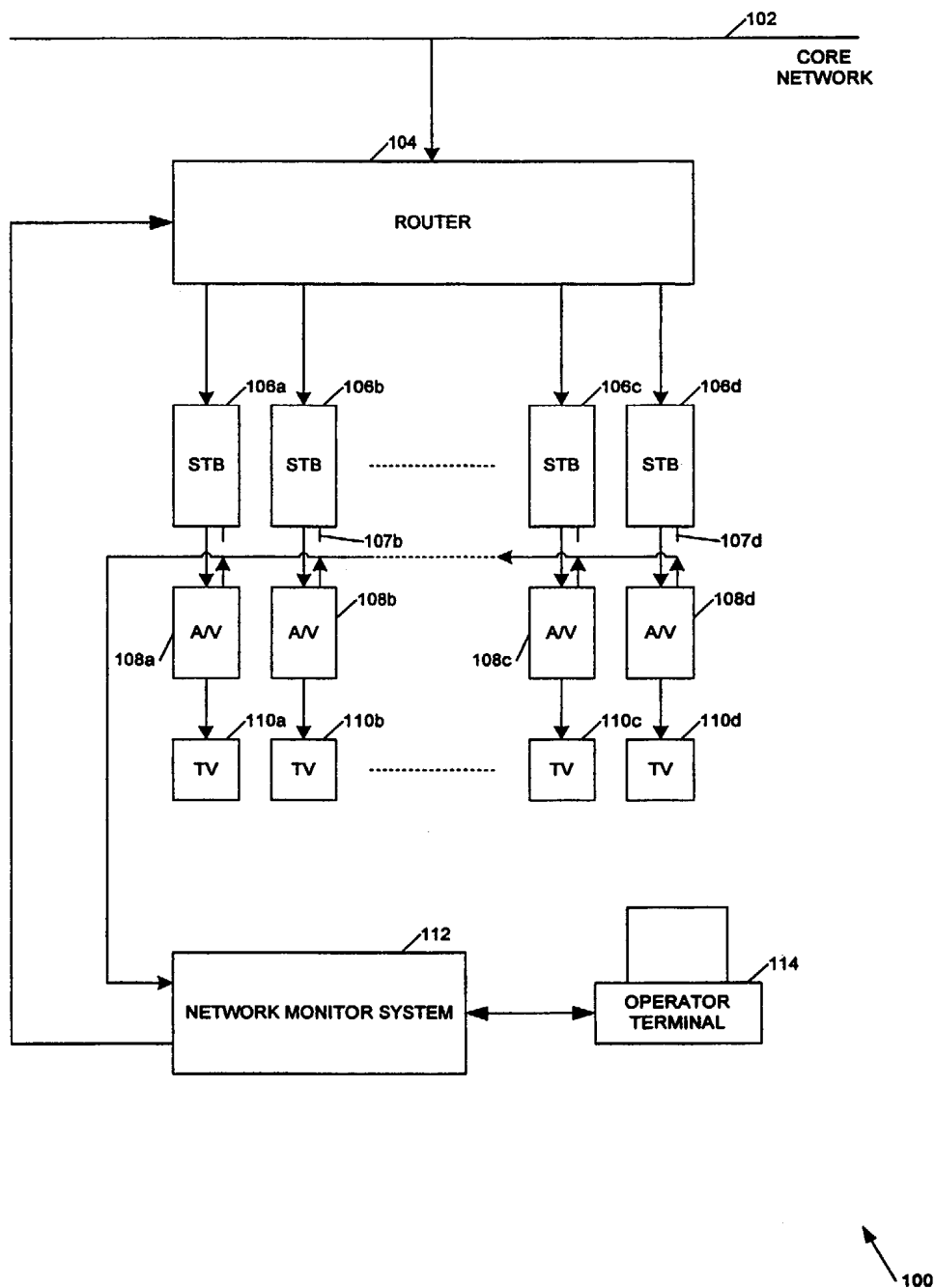


Figure 3



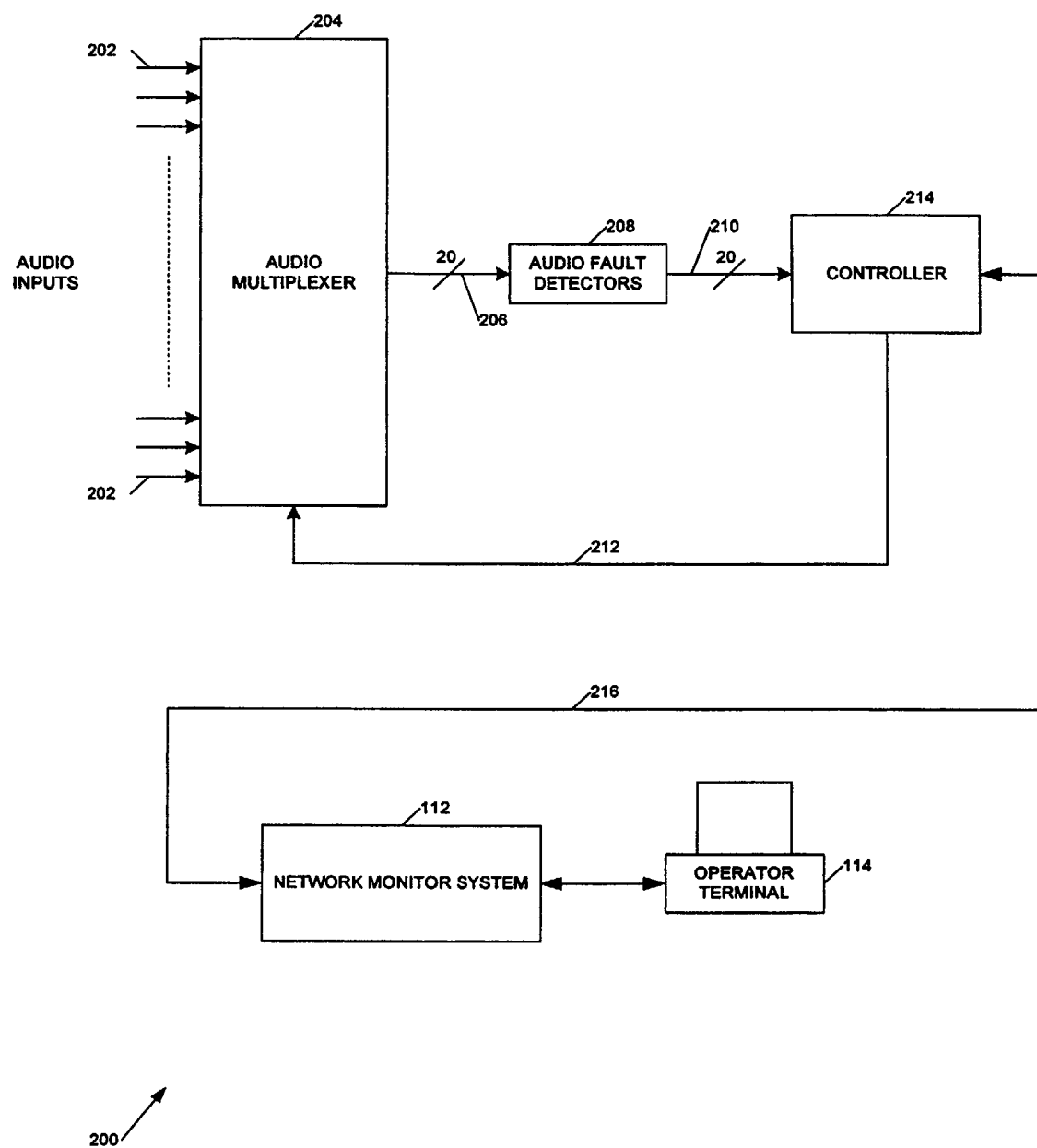


Figure 2

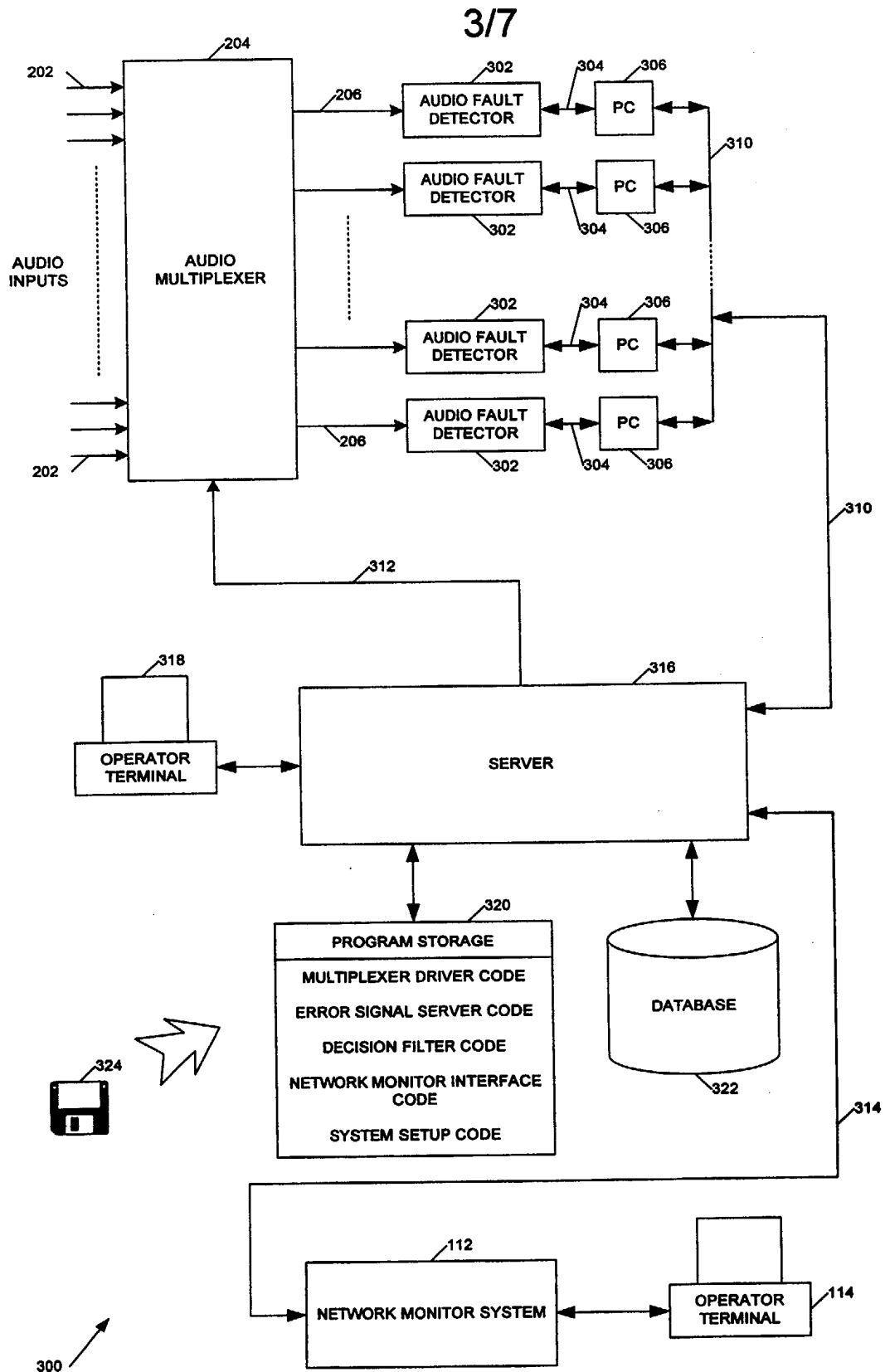


Figure 3

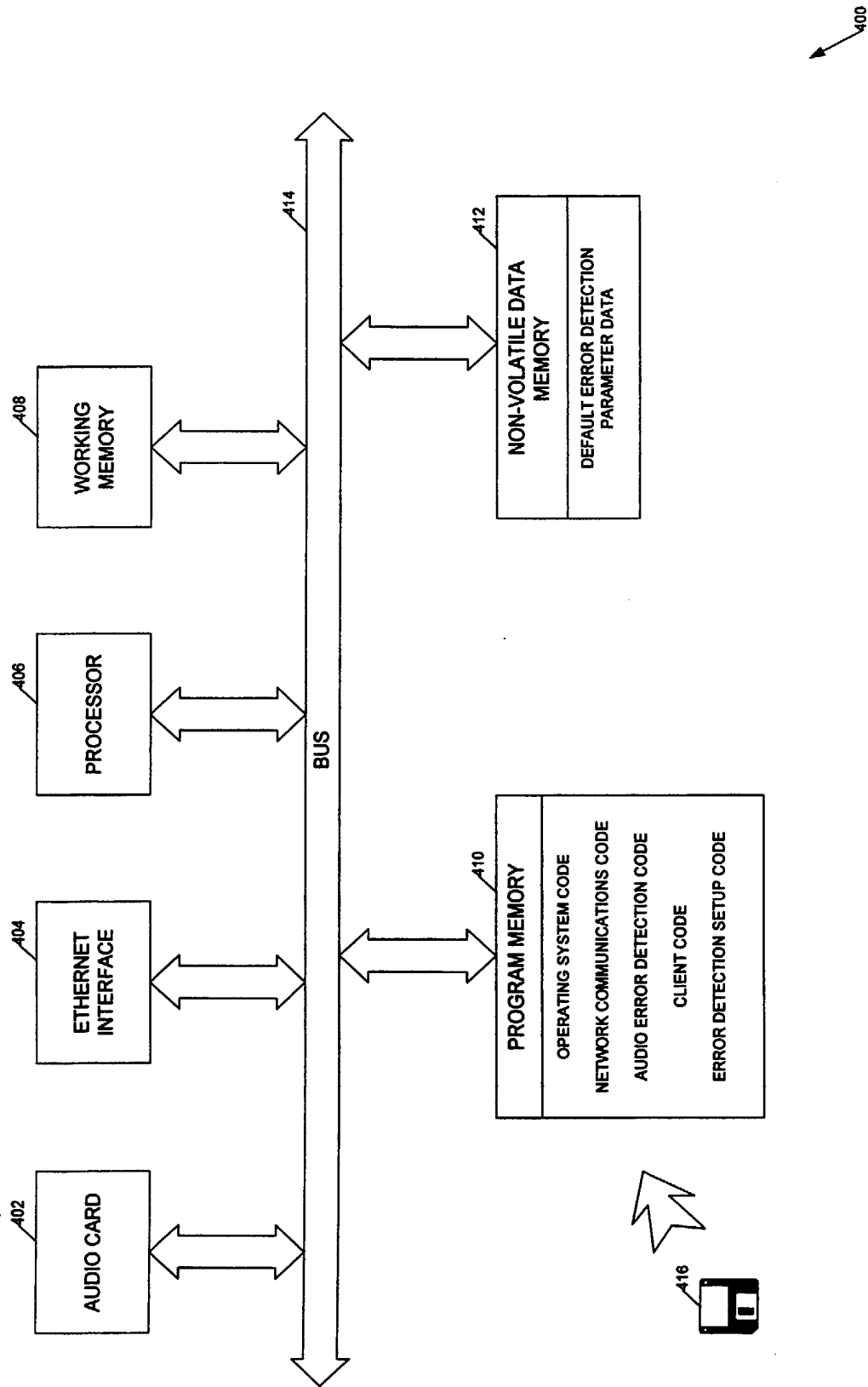


Figure 4

5/7

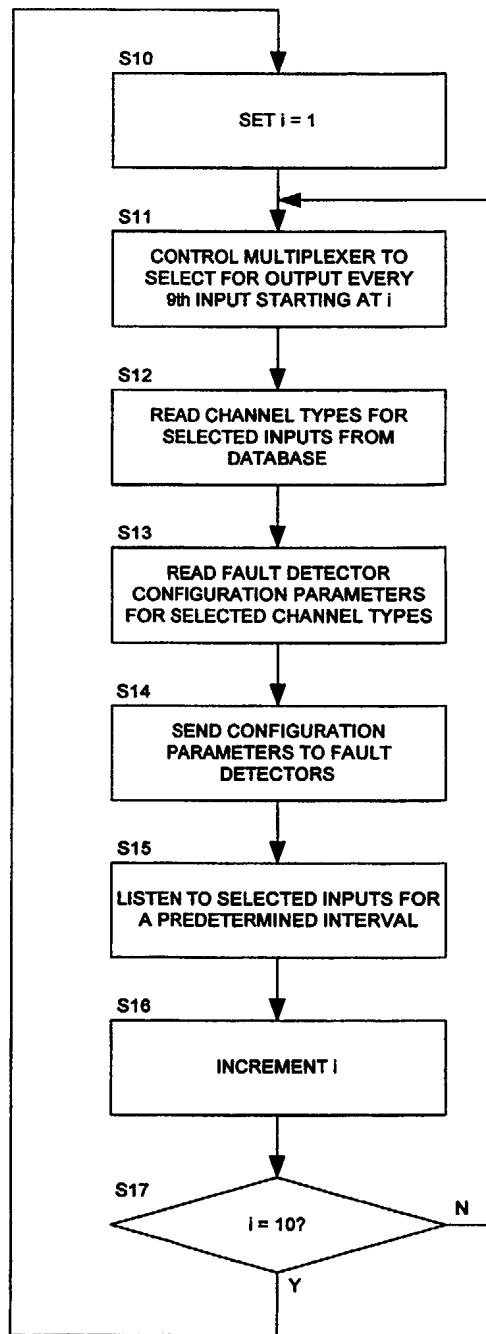


Figure 5

6/7

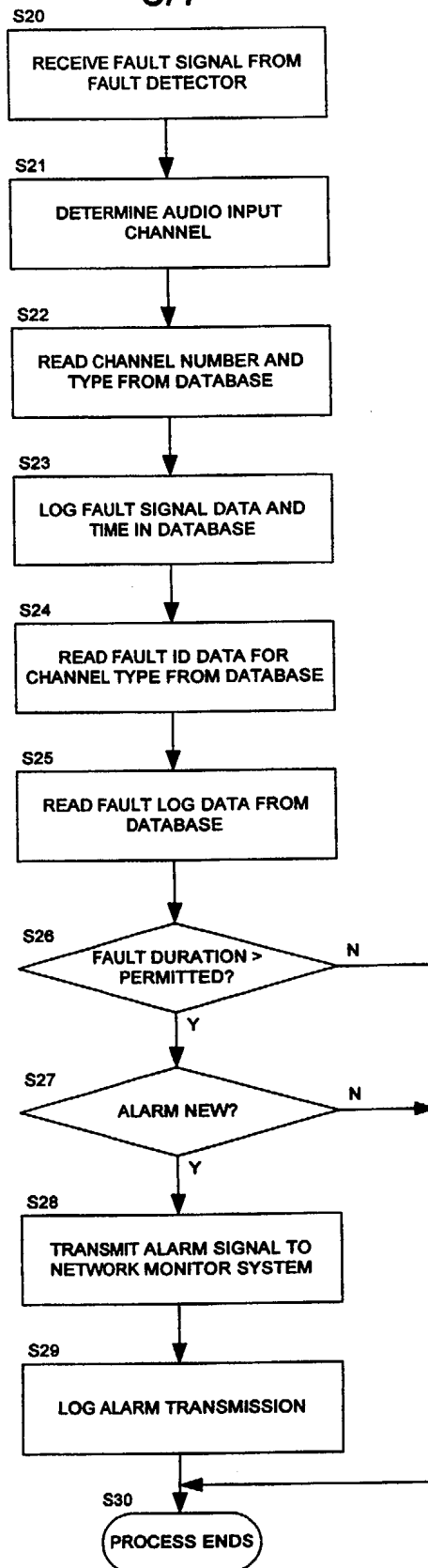


Figure 6

Channels		Templates	
Channel No	Channel Name	Template	
1	Terrestrial	default	
2	Cable 1	default	
3	Cable 2	default	
4	News 1	news 1	
5	News 2	news 2	
6	Film	default	
7	Nature	quiet channel	

700 ↗

702 704 706

Figure 7a

Channels		Templates				
Template	dropout	silence	whitenoise	dropout time	silence time	whitenoise time
default	102	65	30	40	50	60
news 1	66	99	11	12	13	0
news 2	22	33	44	77	44	66
quiet channel	40	202	80	23	99	80

750 ↗

752 754 756 758 760 762 764

Figure 7b

SIGNAL PROCESSING SYSTEMS

This invention is generally concerned with signal processing apparatus and methods for identifying signal faults. It is particularly concerned with the identification of faults in systems with a plurality of audio channels, such as television signal distribution systems.

A cable tv distribution system typically distributes signals for at least 100 different tv channels. The growth in digital encoding of tv signals and in the provision of other services such as interactive multi-media services will increase the number of channels provided by a cable tv system to several hundred, and possibly thousands, of channels. Each of these channels should, ideally, have its audio and video quality monitored.

Presently one approach to the monitoring of tv signals distributed on a cable tv network is to provide a so-called digital media centre (DMC) which acts as a dummy hub on the cable tv network and effectively simulates a plurality of domestic cable tv connections, each tuned to a different one of the channels, but all co-located within the DMC. Such an arrangement is shown in Figure 1.

Referring to Figure 1, a known cable tv signal monitoring system 100 comprises a core cable tv distribution network 102 coupled to a router 104 with a plurality of outputs, one per channel to be monitored. The outputs of router 104 provide a co-axial cable connection carrying analogue or digital tv data (the signal routing/switch equipment in Figure 1 has been simplified for the purposes of illustration). Each output of router 104 is coupled to a set top box 106 a-d which provides an output to a television monitor 110 a-d. The tv monitors 110 are physically arranged as a tv wall, thus allowing a single operator to simultaneously monitor broadcast picture quality on a plurality of channels.

The set top boxes 106 and tv monitors 110 effectively simulate a domestic viewing environment.

Between each set top box 106 and tv monitor 110 is connected an audio-to-video VU meter 108 a-d which provides basic audio monitoring of the tv signal from set top boxes 106. The audio-to-video VU meters 108 (e.g. from Chromatec, Kent, UK; www.chromatec.com) superimpose on the pictures presented on tv monitors 110 a stereo VU meter to allow visual monitoring of sound associated with each tv channel. The audio-to-video VU meters 108 also each provide a "silence detection" output to a network monitor computer system 112. The silence detection outputs provide an alarm signal when the monitored audio remains below a threshold level for a predetermined period of time. The network monitoring system 112 controls router 104 to select channels in use by the cable tv distribution system for monitoring. The network monitoring system 112 is also provided with a terminal 114 for the system operator, to provide silence detection alerts and to facilitate other cable tv network monitoring functions. An exemplary monitoring system is known as "Netcool", and is available from Micromuse Ltd, London, UK (www.micromuse.co.uk). This system provides a range of network monitoring facilities including, for example, nation-wide cable fault monitoring and tracing facilities.

This known cable tv network monitoring system suffers from a number of problems. It is difficult for a single operator to quickly and reliably detect an audio fault on one of the monitored channels by visual inspection of the video VU meters displayed on monitors 110. Although the silence detection alarms assist an operator in detecting when an audio connection has been broken, in practice it is difficult to achieve a useful balance between false alarm signals and missed alarm events. One reason for this is that the audio content of different channels may vary considerably. For example, whereas music or children's tv channels may have virtually continuous high levels of audio, other types or genre of channel, such as film or nature channels, may have relatively long intervals with little or no audio present. One result of this is that the system may be slow to detect audio faults.

A related problem arises in connection with audio faults which do not simply result in complete loss of an audio signal. For example, white (or other coloured) noise may be generated when a device in the broadcast chain becomes de-tuned or unplugged.

Temporary signal dropouts may be caused by weather conditions, such as heavy rain interfering with a microwave or satellite link, and in a digital transmission system, a fault may result in a tone, for example, when a short section of digitised music or speech is trapped within a corrupted buffer and transmitted in a repeating loop. These types of fault have generally been ignored because it is normally even more difficult to separate genuine fault events from an audio stream which could contain intentional tones, white noise, short periods of silence and the like.

Other problems also arise in the above described system, relating to the large total number of channels to be monitored. The detection of audio faults is processor-intensive and the equipment is often bulky and expensive. Providing audio monitoring equipment for each separate channel, when there may be several hundred channels to monitor, is expensive and requires a significant amount of space. Furthermore, the wiring for such a system is also bulky and difficult to install.

These and other problems are addressed by the present invention.

According to a first aspect of the invention there is therefore provided an audio signal processing system for detecting a fault in an input channel, the apparatus comprising an error detector to receive the input channel and to provide an error signal output; and a fault evaluator to receive the error signal and to provide a fault detected output; the fault evaluator including a data store to store channel type data and corresponding fault specification data for each channel type, the channel type data identifying a category of input channel; a program store storing evaluator program code; and a processor coupled to the data store, and to the program store for implementing the program code in the program store, the evaluator program code comprising code to: access channel type data for the input channel and to read corresponding fault specification data from the data

store; and evaluate error signals from the error detector, using the fault specification data from the data store, to detect a fault in the input channel.

The fault evaluator, broadly speaking, is programmed with the type or genre of audio channel being monitored and uses this information to filter signals from the error detector accordingly. The channel type data may be input by an operator or may be read from a database or from an external source, for example, over a network. The channel type data links to monitoring or filtering parameters for the fault evaluator specifying when an audio error or a combination of audio errors is to be treated as an audio fault and when audio "errors" may legitimately be ignored because it is likely that they are part of a properly functioning audio stream for the defined channel type. Channel types may include classical music, pop music, news, nature, film and comedy genres and, preferably, the fault specification data of each genre includes an editable default template. Preferably the signal processing system is an audio signal processing system for detecting a fault in an audio input channel, preferably a stereo input channel, and the error detector is an audio, preferably stereo, error detector.

In a preferred embodiment, a separate set of fault specification data is provided for each type of audio error detectable by the audio error detector. This allows separate filtering processes to be applied to different error types. As described above, error types may include tone errors, white noise (including other noise colours), dropout and silence and each of these error types has a corresponding fault specification template for each channel type into which the audio channels are categorised. In one embodiment a fault is detected by comparing the duration of an error with an error duration threshold value determined by the fault specification data.

As well as filtering the different audio errors based upon channel type, in a preferred embodiment the error detector is itself configured according to the audio channel type of the audio channel it is processing. Thus when the error detector is processing, for example, pop music to detect tone faults, error detection configuration data specific to detecting tone faults in pop music can be downloaded to the detector. Likewise, other

specific error detection data can be downloaded to the error detector for other channel types. This provides a second layer of filtering for the processing apparatus and helps to reduce the number of false positives from the fault detection system as a whole. In one embodiment an error detector is configured to compare a number of error events in a time window with an event threshold value, and to provide an error signal output when the number of events exceeds the threshold value. Thus, effectively, a running total is kept of the number of potential errors within the time window, and this is compared with a significance level. This type of significance threshold can be readily understood by a user.

A preferred embodiment of the system operates to detect faults on a plurality of input channels, using a plurality of error detectors each coupled to the fault evaluator. In this embodiment a data store allocates a channel type to each monitored input channel. Where faults on a plurality of channels are detected, the error detectors may be configured to provide input level data to the fault evaluator, for detecting cross channel level faults, for example, where the audio volume on one channel is significantly greater or less than the audio volume on another channel. The input channels may be multiplexed to select successive channels for coupling to the error detectors for monitoring on a time domain multiple access (TDMA) basis, to reduce the total number of error detectors required.

In a preferred embodiment, an error detector includes a data processor, to simplify interfacing to a plurality of such error detectors. In such an embodiment, the error detectors and the fault evaluator are preferably each coupled to a computer network, such as an ethernet network. This facilitates operation of the system and reduces the wiring demands, particularly where a large number of input channels is monitored.

In another aspect the invention provides a fault detection system for detecting a fault on one of a plurality of input lines, the system comprising a plurality of fault detectors, each having a signal input line for monitoring an input signal for one or more faults and a fault output line for outputting a fault signal to indicate detection of a fault; a data

communications network; a server coupled to the network; and a plurality of data processors, each coupled to the fault output of a respective fault detector and to the network; wherein each data processor includes client monitoring software responsive to a fault signal from the fault detector to which the data processor is coupled to send a fault message over the network to the server, the fault message including data identifying the fault detector which outputted the fault signal; and wherein the server includes server monitoring software to receive a said fault message from a data processor and to identify an input on which the fault occurred.

Coupling the data processors and server using a data communications network, and sending fault messages which include identification information for a fault detector provides an elegant and synergistic combination of features with significant advantages. In particular the combination both reduces the wiring requirements for a system monitoring a plurality of channels and takes advantage of a feature of network communications to solve the problem, which would otherwise arise, of linking a fault detected signal with a channel on which the fault occurs.

In one embodiment, the network is an ethernet network and the identification data comprises, or is derived from, an ethernet media access control (MAC) address or, at a higher layer in the network model, an internet protocol (IP) address. Effectively the source address for the network message from a fault detector identifies the detector, and hence the monitored input line.

In practice where a single fault detector is coupled to each data processor, the fault message sent over the network may identify the data processor rather than the fault detector directly. Where more than one input line is coupled to a fault detector the fault message may identify an input line on which a fault has been detected, rather than identifying the fault detector per se - functionally the system operates to identify an input on which a fault occurs. The fault detector output may comprise an entry in a fault detection log or a signal to another system, such as the network monitor system 112 of Figure 1.

As will be understood by the skilled person, the server may comprise a computer program running on a dedicated machine or it may comprise one of several programs implemented by a machine, and may thus reside on one of the data processors coupled to the network and to a fault detector. The skilled person will also recognise that the fault detectors may be combined with the data processors, for example, by providing a conventional computer with an audio and/or digital signal processor (DSP) card and appropriate fault detection software. In general in a computer network processing functions may be distributed over the network, for example, to optimise the use of resources coupled to the network.

In a preferred embodiment of the system a multiplexer is used to couple a plurality of input channels to selected ones of the fault detectors, preferably under control of the server, either directly or over the network. Thus for example, if there are n input channels for every fault detector, the multiplexer can be used to present each of the n channels allocated to a given fault detector in turn or in rotation to that detector. Since the multiplexer is controlled by the server, the server knows which channel is being monitored when a fault occurs and can thus identify the input channel on which the fault occurs.

The system described above, preferably includes a data store storing data for configuring the fault detectors and storing channel type data for looking up fault filtering data, for filtering the detectors' fault signal outputs to reduce the risk of false alarms. The data store may be coupled directly to the server or to one of the data processors, or may be coupled to the network. The parameters for a fault detector may be accessed by channel type data or, more directly, by a channel number or identifier. Where each channel has an associated set of detector parameters, in a multiplexed system parameters for the channel, rather than for the detector, are written to the detector. In a non-multiplexed system, or where only channels of a specified type are coupled to each detector, configuration data may be stored for each detector rather than for each separate channel.

The fault filtering data may, optionally, be selected according to the fault type being filtered. The filtering software, in one embodiment, checks or selects faults meeting predetermined criteria, for example, faults lasting longer than a predetermined time interval. The filtered fault detection output may comprise a signal transmission or a fault alarm output, for example, to a user or to other apparatus, and/or data written to a data file to log the occurrence of the fault.

In a preferred embodiment, the data communications network is an ethernet network, such as an IEEE 802.3 network or the IEEE compatible DEC/Intel/Xerox Ethernet II network. Preferably the fault detection system is an audio fault detection system and the fault detectors comprise audio fault detectors. Preferably, each audio fault detector monitors a stereo audio channel.

In a further aspect of the system the invention provides an audio fault detection system comprising a plurality of audio inputs for receiving inputs from a plurality of audio channels; an audio signal multiplexer coupled to the plurality of audio inputs and having at least one audio output; at least one audio fault detector having an input coupled to said at least one audio output of the multiplexer and having a fault detection output to indicate the detection of an audio fault; and a controller coupled to the multiplexer to control the multiplexer to select said audio inputs in turn, and coupled to the fault detector to monitor the audio fault detection output of the audio fault detector, and to output a fault signal in response to the fault detection output indicating the detection of an audio fault, the fault signal including an indication of which audio channel has a fault.

In a further aspect there is provided an audio fault detector comprising: a data memory operable to store data to be processed; an instruction memory storing processor implementable program code; a processor coupled to the data memory and to the instruction memory for processing data in accordance with program code stored in the instruction memory; an audio input for an audio input signal; and an analogue-to-digital converter coupled to the audio input to the processor for converting an audio input

signal to a digital signal for processing by the processor; and wherein the instructions stored in the instruction memory comprise instructions for controlling the processor to: capture a sample of the audio input signal; analyse the captured sample; and output an audio fault detected signal in response to the result of said analysis. The fault detector is characterized in that said audio fault comprises a broadband noise fault, for example white noise or tone fault, and said analysis comprises frequency decomposition of the captured sample; and/or in that said audio fault comprises a dropout fault, and said analysis comprises a comparison of levels of the captured audio signal at different times; and/or in that said audio fault comprises a silence fault and said analysis comprises a comparison of a level of the captured audio signal with a threshold level.

In a still further aspect of the system the invention provides a method of detecting a fault in an audio signal comprising determining a fault evaluation category for the audio signal; monitoring the audio signal to detect at least one audio error; reading fault evaluation data from a database corresponding to the fault evaluation category of the signal; and evaluating the at least one audio error, using the fault evaluation data, to detect a fault in the audio signal.

In another aspect the invention provides a method of notification of a fault in one of a plurality of monitored channels, the method comprising inputting the monitored channels in parallel to a plurality of fault detectors; sending a signal from a said fault detector over a computer network to a server on detection of a fault in a channel monitored by the detector, the signal including a source address for the fault detector; and storing or outputting notification data for the fault, the notification data indicating the channel on which the fault occurred.

These and other aspects of the invention will now be further described, by way of example only, with reference to the accompanying figures in which:

Figure 1 shows a known cable tv distribution network monitoring system;

Figure 2 shows a block diagram of a multiplexed audio monitoring system;

Figure 3 shows details of the multiplexed audio monitoring system of Figure 2;

Figure 4 shows an embodiment of a data processor for the system of Figure 3;

Figure 5 shows a flow diagram of a multiplexed audio monitoring process;

Figure 6 shows a flow diagram of a fault detection process; and

Figures 7a and b show exemplary graphical user interfaces for channel fault detection template selection and definition.

Referring again to Figure 1, each set top box 106 has an audio output 107, normally a stereo audio output. This provides a convenient access point for monitoring the audio information associated with the tv channels displayed on tv monitors 110.

Referring now to Figure 2, each audio output 107 is connected to an input 202 of an audio multiplexer 204. In one embodiment audio multiplexer 204 has 192 audio input channels of which 180 are used, and 32 audio output channels of which 20 are used. The audio multiplexer 204 selectively couples the audio inputs to the audio outputs according to signals on a control line 212. The 20 audio outputs 206 are coupled to 20 audio fault detectors 208. The audio multiplexer 204 is controlled to provide time division multiplexing of the audio inputs 202 so that each monitored audio input is coupled, for a predetermined time interval, to an audio fault detector 208. In the embodiment described above with 180 input channels the input channels are grouped into sets of nine channels, and within a group each input is connected in turn to an allotted audio output 206.

The audio multiplexer 204 may comprise a plurality of cascaded multiplexers. Thus, in one embodiment, multiplexer 204 comprises a set of stereo audio matrixes, two

matrixes each having 96 inputs and 32 outputs to provide the 192 audio input channels, each of these matrixes feeding a 64 input 32 output matrix. Such devices are available from, for example, Pro.Bel Limited, Reading, UK and Michael Stevens and Partners Limited, Bromley, UK. A Pro.Bel 6170 matrix controller provides an RS422 interface for control line 212, for interfacing the multiplexer 204 to a conventional computer system.

An audio fault detector 208 may comprise off-the-shelf software running on a computer having an audio input, such as the Wave Corrector software available from Ganymede Test & Measurement (www.ganymede.hemscott.net). Alternatively each audio fault detector 208 may comprise a broadcast audio silence monitor, such as the Model 1100 silence monitor available from Bel. However, preferably the detectors are configured to respond to four audio error types, a single frequency tone, white noise, audio dropout, and silence.

In one embodiment a fault detector comprises a computer with a (stereo) audio input to a sound card on which resides a digital signal processor running processor code, preferably stored within the computer. A single frequency tone error is detected by decomposition of the input signal using a fast Fourier transform (FFT) applied to an audio input time window, and by statistical analysis of audio energy in the frequency domain, in a simple embodiment looking for a high energy narrowband peak. White noise may similarly be detected using analysis, such as statistical analysis, of a frequency decomposition, for example for looking for a uniform broadband signal. Dropout errors may be detected using a weighted comparison of present and historical volume levels with reference to a user-defined threshold. Silence may be detected by comparing an audio input volume level to a second user-defined threshold. The thresholds and statistical significance values may be selected experimentally according to characteristics determined to be approximately representative of the audio channel types to be monitored.

Each fault type detected is therefore preferably associated with one or more parameters determining when an audio fault detected output is to be provided. For example, in the case of tone detection the bandwidth required and energy within that bandwidth may be defined by tone fault parameters and in the case of white noise a minimum signal level required across a defined range of frequencies may be defined for signalling a white noise fault output. In one embodiment an audio fault monitoring process comprises capturing a time slice of audio data by, for example, recording until an input buffer is full, subjecting the recorded time slice to tone, white noise, drop-out and silence fault analysis and, after each analysis, checking against the stored parameters whether or not a fault of the appropriate type should be signalled. The process then loops back to capture a subsequent time slice. Preferably the audio fault analysis code operates sufficiently fast that all four analyses can be performed without missing any input data, that is, in less than the time taken to fill the input buffer. Suitable fast Fourier transform modules are commercially available in, for example, libraries of DSP (digital signal processor) routines.

The audio fault detectors 208 provide outputs 210 to a fault detection system controller 214. Controller 214 also provides control signals on line 212 for audio multiplexer 204, and a fault detected alarm output on line 216 to network monitor system 112. This structure allows a relatively large number of audio input channels to be monitored without requiring a correspondingly large number of audio fault detectors.

The system operates at two levels: firstly the audio input channels are monitored on a time multiplexed-basis, to reduce the number of audio fault detectors required. Secondly, controller 214 provides an additional layer of audio fault detection filtering, reducing the fault detection processing load on detectors 208, because the relatively low bandwidth of signals on line 210 allows a single controller to filter outputs from most, or preferably all, of the audio fault detectors. Furthermore, the audio multiplexer selects audio inputs for time multiplexed processing by the audio fault detectors under the control of controller 214. Thus, when an audio fault is detected, controller 214 knows

which audio input was being monitored by the fault detector issuing the fault signal when the fault signal was generated.

When a fault detector 208 detects a potential audio fault or error, it outputs a signal on one of lines 210 to controller 214, which then evaluates or filters the potential fault to determine whether to signal a fault alarm. In one embodiment a running total of the number of potential errors of each type (tone, white noise, dropout, silence etc.) within a predetermined time window is calculated, and this figure is compared against a user defined threshold for each error type to signal an alarm when the number of potential errors exceeds the relevant threshold. This use of threshold values for fault evaluation/filtering provides a practical compromise between a readily understandable error significance, which is useful for engineers setting up the system, and reliable audio fault detection. The concept of counting potential errors within a window and comparing the count with a preset significance level or threshold can be applied to other error types, in addition to those defined above.

In a preferred embodiment, the audio fault detectors 208 also provide stereo signal level information to controller 214 on output lines 210, to enable the controller to make cross channel level checks on the audio input channels. More specifically, in one embodiment the controller signals a cross channel level fault when the audio level of one channel differs from an average audio level by more than a predetermined threshold value.

The fault alarm outputs from controller 214 are processed by network monitor system 112 and are used to provide a warning to an operator on operator terminal 114. The skilled person will appreciate that a range of operator warning signals may be employed. Thus, for example, terminal 114 may display a window indicating a channel on which a fault has occurred and corresponding fault type (tone, white noise, dropout etc.). In other embodiments, the operator terminal may display an illuminated red/green indicator for each channel to indicate satisfactory operation or a fault.

Preferably operator terminal 114 (or another similar terminal, for example, a terminal connected directly to controller 214) also allows access to system configuration software for configuring the filtering in controller 214 and, optionally, error detection parameters of audio fault detectors 208. Such configuration software allows the user-defined fault alarm threshold referred to above to be defined for each error type and, optionally, for each input channel. The configuration software may also be used to specify channel type data as described in more detail below.

The operator terminal may also include software for low-level monitoring of a selected audio channel, for example, to display the running total of errors or error types on a given channel in a selected time window, and/or frequency decomposition data generated by FFT routines in the audio fault detectors. In a preferred embodiment, controller stores a log of outputs from audio fault detectors 208 which can be read using operator terminal 114 and, preferably, network monitor system 112 stores a log of fault alarm data which can be similarly be displayed and otherwise investigated by an operator using terminal 114. Preferably operator terminal 114 also permits a set up engineer or system user to switch on and off the detection of specified audio error types on selected audio input channels or on selected audio input channel types.

Operator terminal 114 also allows a monitoring engineer to perform other functions such as adding, deleting, or defining an input channel type, adding, deleting or defining a system template (described below), and testing an audio fault detector.

In one embodiment controller 214 comprises a conventional computer system running Windows NT (trade mark) or, preferably, Unix (trade mark) or Linux (trade mark) to simplify interfacing with the Netcool Network Monitor System 112 which runs under Unix. In a preferred embodiment Link 216 comprises an ethernet connection and faults are signalled to the Netcool System using syslog files comprising text messages. In an alternative embodiment controller 214 communicates error messages to the Netcool Network Monitor System 112 using Netcool probe SNMP error and control language.

Referring now to Figure 3, this shows a signal processing system of the type illustrated in Figure 2, in more detail. Each audio fault detector 208 of Figure 2 comprises an audio fault detector 302 coupled via links 304 to a computer 306. In a preferred embodiment, computer 306 is a single board computer system and an audio fault detector comprises a stereo sound card coupled to the computer's bus 304 and audio signal processing software running on computer 306. In the arrangement of Figure 3, the controller 214 comprises a server 316 coupled to programme storage 320 and a database 322 and, optionally to an additional operator terminal 318. Operator terminal 318 provides system and audio fault detection and investigation functions corresponding to those described above with reference to operator terminal 114. An RS 422 output line 312 from server 316 controls audio multiplexer 204. The server 316 is coupled to the network monitor system 112 by means of ethernet link 314. The server 316 and each personal computer (PC) 306 is provided with an ethernet interface to ethernet 310 which couples the personal computer 306 to the server 316.

Programme storage 320 includes multiplexer driver code for controlling audio multiplexer 204 to selectively provide audio input signals to audio fault detectors 302. Programme storage 320 also includes error signal server code for handling the reception of error signals sent by PC 306 over ethernet 310 and decision filter code for filtering or evaluating the error signals in accordance with fault specification or evaluation data stored in database 322. Storage 320 further includes network monitor interface code for interfacing to network monitor system 112 and system set up code for storing and/or modifying data stored in database 322 and for writing audio fault detector configuration data to PCs 306 for configuring audio fault detectors 302 and performing other system set-up functions.

Database 322 stores, for each audio input 202, an audio input number, a channel number, one or more channel descriptors, and channel type data. This allows, for example, input one of audio multiplexer 204 to be defined as channel 57 corresponding to "set top box preset 41", called "film channel 12", having a channel type of "film".

In one embodiment the channel type genres comprise (popular/classical) music, news, documentary, nature, film, cartoon and comedy genres. Such channel type data may be referred to as template data, each channel having a corresponding named template selected from a set of templates set up by an engineer. A channel template stores data for use by server 316 such as threshold values for each audio fault type, and preferably also stores client (audio fault detector) parameters for each channel type.

Thus database 322 preferably also stores fault specification/evaluation data for each fault type (white noise, dropout etc.) for each channel type. This fault specification data is used to evaluate audio error signals sent from audio fault detectors 302 to server 316, in one embodiment using a significance level associated with a running total of potential errors within a time window as described above.

Optionally database 322 also stores configuration parameters for audio fault detectors 302 to configure each audio fault detector for detecting audio faults in a channel of the type at that time connected to the audio fault detector. The multiplexer driver code also then configures each audio fault detector according to the channel type of the channel connected to the audio fault detector at any given time in the time multiplexed input sequence of channel monitoring. In a simplified embodiment, where this optionally audio fault detector parameter set up system is employed, the audio input 202 may be arranged so that only channels of a particular type map to a given audio fault detector, so that an audio fault detector need not be reconfigured each time the audio multiplexer 204 is controlled to present a different input channel to the fault detector.

In an alternative embodiment fault specification/evaluation data is associated directly with each audio input or input channel number, thus avoiding the need to categorise the input channels into specific types. However, in this embodiment fault specification/evaluation data must be set up for each channel individually, which is potentially time consuming. In general, the data in database 322 is set up by a system engineer using operator terminal 318 in an initial system configuration process.

As described above, PCs 306 and server 316 are linked using ethernet 310. In a preferred embodiment TCP/IP (Transmission Control Protocol/Internet Protocol) is used for network communications. The transmission control protocol (TCP) provides a header for each datagram comprising, inter alia, a source and destination port number and a datagram sequence number. At the Internet Protocol level a further header is added specifying source and destination Internet addresses. At the ethernet level an ethernet header is added which includes ethernet source and destination addresses known as media access control (MAC) addresses. Ethernet MAC addresses are registered with a central authority to ensure that all ethernet devices have different addresses. An address resolution protocol (ARP) table constructed by the network links the ethernet MAC addresses to a TCP/IP addresses.

In general a connection is described by both an Internet address and a TCP port number for both the source and destination devices. A combination of a port number and an Internet address is often known as a socket address and, usually, the TCP port number links to an application programme which is listening for messages intended for that ports so that a port number in effect specifies a required service. When a connection is established between a pair of sockets, one of the sockets, at the server, listens for a connection request and the other, at the client, requests the connection. Once two sockets are connected, communication between each client and server is enabled. In most systems some port numbers, typically ports 0 to 1023, are reserved, but other port numbers may be assigned by choice or randomly.

In the present system each PC 306 is assigned a separate socket for communication with server 316, for example, during a system set up process. If more than one audio fault detector 302 is provided per PC 306, each audio fault detector is preferably provided with a unique socket address. Server 316 implements one or more error signal server handling processes to manage audio error/fault signals sent by client software in PCs 306. This provides a convenient method for identifying which audio fault detector and, if necessary, which channel of a stereo input, has a potential audio fault. Each time a fault is detected by an audio fault detector 302, the corresponding PC 306 invokes a

client process to communicate the error/potential fault to server 316. The server then evaluates fault, as described in more detail below with reference to Figure 6.

Referring now to Figure 4, this shows a block diagram of an audio fault detector/PC combination 400, suitable for use with the system of Figure 3. The combination is based on a conventional general purpose computer, suitably programmed, and provided with an audio card 402. The computer also has an ethernet interface 404 for interfacing to ethernet 310, a processor 406, working memory 408, non-volatile data memory 412 for storing default error detection configuration parameters, and programme memory 410 all interconnected by data and communications bus 414.

The non-volatile data memory 412 may comprise non-volatile RAM or ROM and stores default parameters for error detection, such as user defined threshold levels. When other error detection parameters have been downloaded from server 316 to working memory 408, these are used in preference to the default parameters.

The programme memory 410 includes operating system code, such as Windows NT (trade mark) or Windows CE (trade mark), network communications code, audio error detection code for detection of audio errors in accordance with the process as described above, and error signal transmission client code for transmitting audio faults/error signals over ethernet 310 to server error signal code in programme storage 320 of Figure 3. Programme memory 410 also includes error detection set-up code for receiving error detection parameters over ethernet 310 and for configuring the error detection code to operate according to the received parameters.

The code in programme storage 320 and the data in database 322 may be stored on a conventional data storage media such as floppy disk 324 of Figure 3 and, similarly, the code in programme memory 410 and data in non-volatile data memory 412 and/or working memory 408 may be stored on data storage media such as floppy disk 416 of Figure 4.

Figure 5 shows a flow chart of a monitoring process for monitoring signals on the audio inputs 202 of Figures 2 and 3. At step S10 a count variable i is set to 1 and then, at step S11, server 316 controls multiplexer 204 to select every n th input, starting at i , for output to audio fault detectors 302. Where multiplexer 204 has 180 inputs and 20 outputs $n=9$; in general $n=(\text{number of audio inputs}/\text{number of audio outputs})$. Thus inputs $i, n+i, 2n+i, \dots$ are selected for time multiplexed presentation to the audio fault detectors and, at step S12, server 316 looks up in database 322 the channel type of each audio input selected for audio monitoring.

Steps S13 and S14 are optional, but preferable. At step S13 server 316 reads fault detector configuration parameters for all the channel types being presented to the audio fault detectors 302. Then, at step S14, these fault detector configuration parameters are written to the audio fault detectors, each detector receiving parameters corresponding to the channel type it is at that time monitoring. Thus the audio fault detectors are optimised for detecting faults on the time multiplexed input channels.

At step S15 fault detectors 302 and, correspondingly, error signal server code processes, in programme storage 320 listen to the selected audio inputs for a predetermined time interval, monitoring for audio errors within the time interval. If an error is detected, the process of Figure 6 is invoked for the monitored channel. In a preferred embodiment a separate program thread is spawned or assigned for each monitored channel, for concurrent channel monitoring.

After the predetermined time interval has lapsed, at step S16 the count i is incremented and, at step S17, tested to check whether it has reached $(n+1)$, in the example, 10. If the count has not yet reached $(n+1)$ the process loops to step S11 to select the next set of inputs for monitoring; if $(n+1)$ has been reached, the process loops to step S10 and the count is reset to 1 to monitor the initial set of audio inputs again. In this way the audio inputs are selected in rotation for monitoring.

In general audio multiplexer 204 will output noise for a brief period whilst the audio inputs are being switched. Thus it is preferable that between steps S11 and S15 of Figure 5 a delay is introduced (not shown in Figure 5) to take account of the channel switching latency, which is generally a known, hardware-dependent time interval. In this way spurious errors generated by audio channel switching are reduced.

In a modification to the process of Figure 5, if the decision filter code in programme storage 320 of Figure 3 detects that an error duration is close to the time threshold at which a fault alarm will be generated, a loop hold signal can be used to temporarily halt the process of Figure 5 at step S15, to check whether the alarm threshold is in fact reached. The process for a halted for a predetermined time interval such as 10, 20 or 30 seconds, defined by, for example, a system engineer during system commissioning. Such a time delay may be interrupted by a new error signal. A similar delay may also be introduced when the decision filter code indicates that a channel may have an intermittent fault.

The loop timing is determined by the total number of audio inputs to be monitored, the number of audio fault detectors, and the minimum acceptable monitoring time for each channel. Thus, the delay between successive monitoring events for an audio input (the loop time) = (number of audio channels to be monitored x acceptable channel monitoring time)/number of audio fault detectors. Alternatively the number of audio fault detectors may be selected based upon an acceptable loop time as the number of fault detectors = (number of audio channels x acceptable channel monitoring time)/acceptable loop time. Thus if, for example, an acceptable loop time is 180 secs and an acceptable channel monitoring time is 20 secs, to monitor 180 channels requires 20 audio fault detectors. In practice the number of audio fault detectors may also depend upon a cost, loop-time trade-off.

Figure 6 shows a flow diagram of a decision filter/fault evaluator code process for determining whether or when a fault alarm should be issued by server 316 when an audio fault detector 302 generates a fault signal.

At step S20, server 316 receives a fault signal from a fault detector and identifies the fault detector sending the signal using the socket address of the fault detector, or PC to which it is connected. This fault signal is received by error signal code running on server 316 and passed to a decision filter code process. The decision filter code process, at step S21, determines the audio input channel on which potential fault occurred, using multiplexer status data indicating which audio inputs 202 are connected to which output 206 of multiplexer 204. Then, at step S22, the decision filter code reads the channel number and type for the channel on which the fault occurred from database 322 and, at step S23, writes a fault log entry into database 322 comprising, inter alia, data identifying the fault type, channel number and the time at which the fault occurred.

At step S24 the decision filter then reads fault specification/evaluation data for the fault type and channel type of the channel on which the fault occurred, to evaluate the fault to determine whether a fault alarm should be generated from the received fault signal. Any suitable fault evaluation algorithm can be used, but in a preferred embodiment one evaluation criterion is the fault duration, as described below with reference to Figure 7.

At step 25 the fault filter/evaluator process reads fault log data from database 322 to determine whether the channel fault has previously been logged, for example, in the immediately preceding loop process of Figure 5. If so, at step S26 the process calculates an approximate fault duration and compares this with a permitted fault duration value specified by the fault evaluation data for that fault and channel type, as read in step S24. As each input is monitored for a predetermined period of time during the monitoring loop of Figure 5, the presence of the fault may also be monitored in real time at step S26, during the monitoring window for the input, to determine whether the fault duration exceeds the permitted threshold. With such an arrangement PCs 306 are configured, for example, to generate repeated messages whilst a fault is present. If, at step S26, the fault duration is approaching the permitted alarm threshold, this step may also generate a loop hold signal to temporarily halt the loop of Figure 5 as described above. Thus, for example, a loop hold signal may be generated if the duration is

approaching the threshold duration and/or if a new fault is detected close to the point in time at which the loop process of Figure 5, would normally switch to a new input channel to be monitored.

If, at step S26, it is determined that the fault duration is less than the permitted alarm threshold, the process ends at step S30 and, if the fault is still present when the input channel is next monitored, this would be detected as step S25 and the fault duration would again be compared with the permitted value.

If the fault duration is greater than the permitted threshold for generating a fault alarm, at step S27 the decision filter code determines whether or not an alarm has already been generated for this fault, by reading from an alarm transmission log in database 322. Again, if the fault has already generated an alarm the process ends at step S30. If an alarm has not yet been generated for the fault the process continues to step S28 and server 316 transmits an alarm signal to network monitor system 112. This alarm signal comprises audio channel identity data such as a channel number, channel description and channel type as well as, preferably, the audio input 202 for the audio channel, and data identifying the fault type. Optionally, other information may also be transmitted with the alarm signal, such as fault duration data and detailed fault information such as error frequency data, audio channel frequency decomposition data and user-defined threshold levels for a audio fault detector for detecting a fault. Finally, at step S29, the fault alarm signal transmission is logged in database 322, preferably together with the other fault-related data transmitted to the network system 112. The fault detection process then ends at step S30.

Referring to Figures 7a and 7b these show, respectively, a graphical user interface 700 for associating an audio fault detection template with each audio input channel, and a graphical user interface 750 for defining such a template.

In Figure 7a field 702 defines a channel number, field 704 a corresponding channel name and field 706 the selected template for the channel. The template is preferably

selected using a drop-down list from one of a set of named, pre-programmed templates such as the "default", "news 1", "news 2", and "quiet channel" templates illustrated.

In the GUI 750 of Figure 7b parameters are displayed for each of the defined templates. The graphical user interface allows the parameter data to be modified and allows the addition of new templates and the deletion of existing templates. In the illustrated embodiment six parameters are defined for each template, although in other embodiments more or fewer parameters may be defined. The name of each template is displayed in a template name field 752. Associated with each template is a pair of parameters 754, 760 for drop-out errors, a pair of parameters 756, 762 for silence errors, and a pair of parameters 758, 764 for white noise errors. Each pair of parameters comprises, in a preferred embodiment, a threshold level 754, 756, 758 for the parameter and a duration 760, 762, 764 for which a fault must be present before an error is signalled.

In general each named template is associated with a different set of threshold and duration values for each audio fault type. Thus in the illustrated example relatively long periods of silence and white noise are required on a "quiet channel" to trigger an error. Similarly the threshold levels for silence and white noise (which are used to set parameters for audio fault detectors 208/302) are relatively high, implying a relatively high level of white noise or a relatively quite level for silence is necessary to trigger an audio fault.

The invention is not limited to the described embodiments and may, for example, be applied to monitoring faults in signals other than audio signals, such as faults in video signals. The invention is not restricted to use in a cable tv monitoring system and can also be used, for example, for monitoring radio broadcasts, web casts, and other transmissions.

Effective alternatives within the spirit and scope of the present invention will occur to those skilled in the art and the invention is not limited to the above described embodiments.

CLAIMS:

1. A signal processing system for detecting a fault in an input channel, the apparatus comprising:
 - an error detector to receive the input channel and to provide an error signal output; and
 - a fault evaluator to receive the error signal and to provide a fault detected output; the fault evaluator including:
 - a data store to store channel type data and corresponding fault specification data for each channel type, the channel type data identifying a category of input channel;
 - a program store storing evaluator program code; and
 - a processor coupled to the data store, and to the program store for implementing the program code in the program store, the evaluator program code comprising code to:
 - access channel type data for the input channel and to read corresponding fault specification data from the data store; and
 - evaluate error signals from the error detector, using the fault specification data from the data store, to detect a fault in the input channel.
2. A signal processing system as claimed in claim 1, wherein the error detector provides an error signal output comprising data indicating one of a plurality of error types, wherein the data stores fault specification data for a plurality of faults, corresponding to the plurality of error types, for each channel type, and wherein the code to read fault specification data reads fault specification data corresponding to a detected error type.
3. A signal processing system as claimed in claim 2, wherein the error detector detects audio error types selected from a group comprising tone, white noise, dropout and silence error types.
4. A signal processing system as claimed in claim 1, 2 or 3, wherein the data store further comprises error detector configuration data corresponding to each channel type,

and wherein the fault evaluator program code further comprises code to read the error detector configuration data corresponding to the input channel type data from the data store and to write the configuration data to the error detector.

5. A signal processing system as claimed in any one of claims 1 to 4, wherein the error detector is configured to compare a number of error events in a time window with an event threshold value and to provide an error signal output when the number of events exceeds the threshold value.
6. A signal processing system as claimed in any preceding claim, wherein the code to evaluate error signals comprises code to compare a duration of an error with a duration threshold value determined by the fault specification data.
7. A signal processing system as claimed in any one of claims 1 to 6, wherein the evaluator program code further comprises code to log an error and/or a detected fault.
8. A signal processing system as claimed in any one of claims 1 to 7, for detecting faults on a plurality of input channels, comprising a plurality of said error detectors each coupled to the fault evaluator, and wherein the data store of the fault evaluator is configured to store channel type data for each said input channel.
9. A signal processing system as claimed in claim 8, wherein each said error detector is configured to provide input level data to the fault evaluator, and wherein the fault evaluator program code further comprises code to detect a cross channel level fault.
10. An signal processing system as claimed in any preceding claim, wherein a said error detector comprises a data processor and an error detector program store storing error detection program code for implementation by the data processor.

11. A signal processing system as claimed in claim 10, wherein said error detector or detectors and the fault evaluator are coupled over a computer network.

12. A fault detection system for detecting a fault on one of a plurality of input lines, the system comprising:

a plurality of fault detectors, each having a signal input line for monitoring an input signal for one or more faults and a fault output line for outputting a fault signal to indicate detection of a fault;

a data communications network;

a server coupled to the network; and

a plurality of data processors, each coupled to the fault output of a respective fault detector and to the network;

wherein each data processor includes client monitoring software responsive to a fault signal from the fault detector to which the data processor is coupled to send a fault message over the network to the server, the fault message including data identifying the fault detector which outputted the fault signal; and

wherein the server includes server monitoring software to receive a said fault message from a data processor and to identify an input on which the fault occurred.

13. A fault detection system as claimed in claim 12, further comprising a multiplexer to receive inputs on a plurality of input channels and to output selected ones of the input channels to the input lines of the fault detectors; and wherein the server is coupled to the multiplexer and further comprises multiplexer software to control the multiplexer to provide selected input channels to the fault detectors and software to identify an input channel on which a fault has occurred.

14. A fault detector system as claimed in claim 13, wherein each fault detector monitors one of a plurality of subsets of the input channels and wherein the multiplexer software controls the multiplexer to provide each one of a said subset of input channels to its corresponding fault detector in rotation.

15. A fault detector system as claimed in claim 13 or 14, wherein the server or each data processor further comprises a data store storing one or more detector parameters for configuring a said fault detector and wherein each data processor further comprises detector configuration software to read the detector parameters from the data store and to write the parameters to a fault detector.

16. A fault detector system as claimed in claim 15, wherein each detector or channel has an associated set of detector parameters, and wherein the detector configuration software further comprises code to look up the detector parameters for a detector or channel for writing to a said fault detector.

17. A fault detector system as claimed in any one of claims 13 to 16, further comprising a fault filtering data store storing fault filtering data for each input channel, and wherein the server software includes filtering software responsive to identification of an input channel on which a fault has occurred to read the fault filtering data for the channel and to provide a filtered fault detection output.

18. A fault detection system as claimed in claim 17, wherein each input channel is categorized into one of a plurality of channel types, wherein the fault filtering data store stores channel type data for each input channel and fault filtering data for each channel type, and wherein, to read the fault filtering data for the channel, the filtering software reads the channel type data for a channel and the fault filtering data for the channel type.

19. A fault detection system as claimed in claim 17 or 18, wherein the fault message from the data processor identifies a fault type, wherein the fault filtering data store stores fault filtering data for each fault type, and wherein the filtering software is responsive to the fault type to read fault filtering data for the fault type.

20. A fault detection system as claimed in claim 17, 18 or 19, wherein the filtering software further comprises code to store fault event data responsive to reception of a

said fault message and code to read said stored fault event data and to provide said filtered fault detection output responsive to said stored fault event data.

21. A fault detection system as claimed in any one of claims 12 to 20, wherein the server monitoring software further comprises code to provide a fault alarm output in response to reception of a said fault message by the server.

22. A fault detection system as claimed in any one of claims 12 to 21, wherein the network is an ethernet network and wherein the data identifying a fault detector comprises or is derived from an ethernet media access control (MAC) address.

23. A signal processing or fault detection system as claimed in any one of claims 1 to 22 for audio fault detection, wherein the fault or error detectors comprise audio fault or error detectors.

24. An audio fault detection system comprising:
a plurality of audio inputs for receiving inputs from a plurality of audio channels;
an audio signal multiplexer coupled to the plurality of audio inputs and having at least one audio output;

at least one audio fault detector having an input coupled to said at least one audio output of the multiplexer and having a fault detection output to indicate the detection of an audio fault; and

a controller coupled to the multiplexer to control the multiplexer to select said audio inputs in turn, and coupled to the fault detector to monitor the audio fault detection output of the audio fault detector, and to output a fault signal in response to the fault detection output indicating the detection of an audio fault, the fault signal including an indication of which audio channel has a fault.

25. An audio fault detection system as claimed in claim 24, wherein said multiplexer has a plurality of outputs, further comprising a plurality of said audio fault detectors, each coupled to a said multiplexer output.

26. A method of detecting a fault in an audio signal, comprising:
determining a fault evaluation category for the audio signal;
monitoring the audio signal to detect at least one audio error;
reading fault evaluation data from a database corresponding to the fault evaluation category of the signal; and
evaluating the at least one audio error, using the fault evaluation data, to detect a fault in the audio signal.
27. A method of providing notification of a fault in one of a plurality of monitored channels, the method comprising:
inputting the monitored channels in parallel to a plurality of fault detectors;
sending a signal from a said fault detector over a computer network to a server on detection of a fault in a channel monitored by the detector, the signal including a source address for the fault detector; and
storing or outputting notification data for the fault, the notification data indicating the channel on which the fault occurred.
28. A method as claimed in claim 27, further comprising:
controlling a selector to sequentially select channels input to the fault detectors from the plurality of monitored channels; and
determining the channel on which the fault occurred using a status of said sequential selection on detection of a fault.



INVESTOR IN PEOPLE

Application No: GB 0104955.0
Claims searched: 1 and 26 only

Examiner: Gareth Lewis
Date of search: 30 October 2001

Patents Act 1977 Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.S): H2K (KHF KJBC KJS KSC2) H4P (PEM)

Int Cl (Ed.7): H04B (17/00) H04H (1/02) H04N (17/00)

Other: Online : WPI JAPIO EPODOC

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	EP 1014279 A1 (XEROX) See abstract.	
A	WO 00/38358 A2 (INTEL) See whole document	
A	US 4752927 (MELLING) See abstract and column 1	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.